

```

enqueue( queue i)
{
    if (enqueue_count == dequeue_count)           // check #1
    {
         $LES_i^N = CDS^N$ ;                          // queue is empty
    }
    else if ((( $CDS^N - LES_i^N$ ) mod N) < M)           // check #2
    {
         $LES_i^N = CDS^N$ ;                          // queue is empty and dequeue count is lagging
    }

     $LES_i^N = (LES_i^N + j)$  mod N;                    // calculate where to enqueue the packet
                                                    // value j < M depends on queuing scheme
                                                    // Note:  $LES_i^N$  increases,  $CDS^N$  unchanged

    if ((( $CDS^N - LES_i^N$ ) mod N) < M)               // check #3
    {
        Drop packet                                // queue has overflowed
         $LES_i^N = (LES_i^N - j)$  mod N;                // reset  $LES_i$  to old value
    }
    else
    {
        Enqueue the packet
    }
}

while (1)
{
    enqueue(i);                                     // call enqueue routine for queue i
                                                    // value  $LES_i$  may increase

    Perform some dequeues
    Perform enqueues on other queues
     $CDS^N = (CDS^N + 1)$  mod N;                        // 1 depends on how many rounds
                                                    // have completed dequeues
                                                    // Note:  $LES_i^N$  unchanged,  $CDS^N$  increases
}

```

FIG. 6